



NT  
VO

```

LL                      IIIII
LL                      IIIII
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LL                      II
LLL LLLLLLLLLL         IIIII
LLL LLLLLLLLLL         IIIII
SSSSSSSS               SS
SSSSSSSS               SS
SS                     SS
SS                     SS
SSSSSS                SSSSSS
SSSSSS                SSSSSS
                        SS
                        SS
                        SS
                        SS
SSSSSSSS             SSSSSSSS
SSSSSSSS             SSSSSSSS

```

|     |     |                                                |
|-----|-----|------------------------------------------------|
| (2) | 65  | DECLARATIONS                                   |
| (3) | 100 | NT\$CONNECT - PERFORM NETWORK CONNECT FUNCTION |
| (4) | 217 | NT\$CHK_SQO                                    |
| (5) | 278 | NT\$CHK_RAC                                    |
| (7) | 354 | NT\$MAP_ROP                                    |

```

0000 1          $BEGIN NTOCONN,000,NF$NETWORK,<NETWORK CONNECT STREAM>
0000 2
0000 3
0000 4
0000 5 *****
0000 6 *****
0000 7 *
0000 8 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 *  ALL RIGHTS RESERVED.
0000 11 *
0000 12 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 *  TRANSFERRED.
0000 18 *
0000 19 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 *  CORPORATION.
0000 22 *
0000 23 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27 *****
0000 28
0000 29 ++
0000 30 Facility: RMS
0000 31
0000 32 Abstract:
0000 33
0000 34 This module communicates with the file access listener at the remote
0000 35 node to establish a record stream for the specified file.
0000 36
0000 37 Environment: VAX/VMS, executive mode
0000 38
0000 39 Author: James A. Krycka,      Creation Date: 09-DEC-1977
0000 40
0000 41 Modified By:
0000 42
0000 43
0000 44 V03-004 JEJ0009      J E Johnson      16-Mar-1984
0000 45 Increase the level of checking on the interaction of the
0000 46 SQO bit and the various RAC options.
0000 47
0000 48 V03-003 JEJ0004      J E Johnson      01-Mar-1984
0000 49 Disable file transfer mode for access of image indirect
0000 50 process permanent files to permit proper network action
0000 51 across various image invocations, as in the handling of
0000 52 $DISPLAY.
0000 53
0000 54 V03-002 JAK0124      J A Krycka      06-SEP-1983
0000 55 Make corresponding source code change for VMS V3.5 patch in
0000 56 support of VAXELAN.
0000 57

```

NTOCONN  
V04-000

NETWORK CONNECT STREAM

J 12

15-SEP-1984 23:51:33 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:21 [RMS.SRC]NTOCONN.MAR;1

Page 2  
(1)

0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :--

V03-001 JAK0123 J A Krycka 29-AUG-1983  
Remove check for a MBX device in NT\$CHK\_S00 to permit use of  
DAP file transfer mode during access to the null device or to  
a mailbox on the remote node.

```

0000 65      .SBTTL  DECLARATIONS
0000 66
0000 67 :
0000 68 : Include Files:
0000 69 :
0000 70
0000 71      $DAPPLGDEF      ; Define DAP prologue symbols
0000 72      $DAPHDRDEF     ; Define DAP message header
0000 73      $DAPCNFDEF     ; Define DAP Configuration message
0000 74      $DAPCTLDEF     ; Define DAP Control message
0000 75      $IFBDEF        ; Define IFAB symbols
0000 76      $IRBDEF        ; Define IRAB symbols
0000 77      $NWADEF        ; Define Network Work Area symbols
0000 78      $RABDEF        ; Define Record Access Block symbols
0000 79      $RMSDEF        ; Define RMS error code symbols
0000 80
0000 81 :
0000 82 : Macros:
0000 83 :
0000 84 :      None
0000 85 :
0000 86 : Equated Symbols:
0000 87 :
0000 88
0000 89      ASSUME  DAP$Q_DCODE_FLG EQ 0
0000 90      ASSUME  NWA$Q_FLG EQ 0
0000 91
00000030 0000 92      PARTNER = DAP$W_PARTNER * 8      ; Offset to PARTNER field
0000 93
0000 94 :
0000 95 : Own Storage:
0000 96 :
0000 97 :      None
0000 98 :

```

```

0000 100      .SBTTL NT$CONNECT - PERFORM NETWORK CONNECT FUNCTION
0000 101
0000 102      :++
0000 103      : NT$CONNECT - engages in a DAP dialogue with the remote FAL to establish a
0000 104      : record stream for the specified file.
0000 105
0000 106      : Calling Sequence:
0000 107
0000 108      :     BSBW      NT$CONNECT
0000 109
0000 110      : Input Parameters:
0000 111
0000 112      :     R8      RAB address
0000 113      :     R9      IRAB address
0000 114      :     R10     IFAB address
0000 115      :     R11     Impure Area address
0000 116
0000 117      : Implicit Inputs:
0000 118
0000 119      :     DAP$Q_SYSCAP bits APPEND, SEQRAC, SEQFIL, SWMODE,
0000 120      :     RANRRN, RANKEY, RANRFA, RANVBN
0000 121      :     DAP$V_GEQ V52, DAP$V_GEQ 54, DAP$V_GEQ_60
0000 122      :     DAP$V_VAXVMS, DAP$V_VAXECAN
0000 123      :     IFB$V_BIO
0000 124      :     IFB$V_BRO
0000 125      :     IFB$V_SQO
0000 126      :     RAB$B_KRF
0000 127      :     RAB$B_RAC
0000 128      :     RAB$L_ROP bits
0000 129
0000 130      : Output Parameters:
0000 131
0000 132      :     R0      Status code (RMS)
0000 133      :     R1-R7   Destroyed
0000 134      :     AP      Destroyed
0000 135
0000 136      : Implicit Outputs:
0000 137
0000 138      :     IFB$V_BIO
0000 139      :     IRB$V_DAP_CONN
0000 140      :     NWA$V_FTM_EOF
0000 141      :     NWA$V_FTM_INIT
0000 142      :     NWA$V_RAC_FLG
0000 143
0000 144      : Completion Codes:
0000 145
0000 146      :     Standard RMS completion codes
0000 147
0000 148      : Side Effects:
0000 149
0000 150      :     None
0000 151
0000 152      :--
0000 153
0000 154      NT$CONNECT::      : Entry point
0000 155      $STPT NTCONN      :
0000 156      MOVL  IFB$L_NWA_PTR(R10),R7 : Get address of NWA (and DAP)
57  3C AA  D0  0006

```

```

10 04 08 E1 000A 157 BBC #RAB$V_BIO,- ; Set internal block I/O flag if
06 22 06 E1 000C 158 RAB$L_ROP(R8),SEND_CTL ; partner supports BRO and user set
06 22 06 E1 000F 159 BBC #IFB$V_BRO,- ; both FAC BRO and ROP BIO bits, thus
06 22 06 E1 0011 160 IFB$B_FAC(R10),SEND_CTL ; declaring block I/O mode on connect
06 22 06 B3 0014 161 BITW #<<102DAP$V_VAXVMS-PARTNER>>!--
06 22 06 B3 0018 162 <102DAP$V_VAXELAN-PARTNER>>!--
06 22 06 B3 0018 163 0>,DAP$W_PARTNER(R7) ; Branch if partner is neither VAX/VMS
06 22 06 B3 0018 164 BEQL SEND_CTL ; nor VAXELAN
06 22 06 B3 001A 165 $SETBIT #IFB$V_BIO,IFB$B_FAC(R10)
06 22 06 B3 001F 166 ;+
06 22 06 B3 001F 167 ; Build and send DAP Control message to partner.
06 22 06 B3 001F 168 ;+
06 22 06 B3 001F 169 ;+
06 22 06 B3 001F 170 ;+
06 22 06 B3 001F 171 SEND_CTL:
06 22 06 B3 001F 172 $SETBIT #NWA$V_LAST_MSG,(R7) ; Declare this last message to block
50 04 04 D0 0023 173 MOVL #DAP$K_CTL_MSG,R0 ; Get message type value
50 04 04 D0 0026 174 BSBW NT$BUILT HEAD ; Construct message header
85 02 02 90 0029 175 MOVB #DAP$K_CONNECT,(R5)+ ; Store CTLFUNC field
85 02 02 90 002C 176 MOVB #DAP$M_ROP,(R5)+ ; Store CTLMENU field
85 02 02 E0 002F 177 BBS #IFB$V_BIO,- ; Branch if block I/O specified
20 0F 22 AA 0031 178 IFB$B_FAC(R10),10$ ;
20 0F 22 AA 0034 179 CMPB NWA$B_ORG(R7),#NWA$K_IDX ; Branch if not IDX organization
20 0F 22 AA 0039 180 BNEQ 10$ ;
20 0F 22 AA 0039 181 BISB2 #DAP$M_KRF,-1(R5) ; Add KRF field to menu
20 0F 22 AA 0039 182 MOVB RAB$B_RRF(R8),(R5)+ ; Store key of reference field
20 0F 22 AA 0039 183 BSBW NT$MAP_ROP ; Store ROP field
20 0F 22 AA 0039 184 BBC #RAB$V_EOF,- ; Branch if position to end-of-file
20 0F 22 AA 0039 185 RAB$L_ROP(R8),20$ ; was not specified
20 0F 22 AA 0039 186 BBC #DAP$V_APPEND,- ; Branch if partner does not support
20 0F 22 AA 0039 187 DAP$Q_SYSCAP(R7),ERRSUP ; file append
20 0F 22 AA 0039 188 20$: BSBW NT$BUILT TAIL ; Finish building message
20 0F 22 AA 0039 189 BSBW NT$TRANSMIT ; Send Control message to FAL
20 0F 22 AA 0039 190 BLBC R0,EXIT ; Branch on failure
20 0F 22 AA 0039 191
20 0F 22 AA 0039 192 ;+
20 0F 22 AA 0039 193 ; Receive DAP Acknowledge message from partner.
20 0F 22 AA 0039 194 ;+
20 0F 22 AA 0039 195 ;+
20 0F 22 AA 0039 196 RECV_ACK:
20 0F 22 AA 0039 197 $SETBIT #DAP$K_ACK_MSG,DAP$L_MSG_MASK(R7) ; Expect response of Acknowledge message
20 0F 22 AA 0039 198 BSBW NT$RECEIVE ; Get reply from FAL
20 0F 22 AA 0039 199 BLBC R0,EXIT ; Branch on failure
20 0F 22 AA 0039 200
20 0F 22 AA 0039 201 ;+
20 0F 22 AA 0039 202 ; Perform initialization functions.
20 0F 22 AA 0039 203 ;+
20 0F 22 AA 0039 204 ;+
20 0F 22 AA 0039 205 ;+
20 0F 22 AA 0039 206 $SETBIT #IRB$V_DAP_CONN,(R9) ; Yes, FAL has completed connect
20 0F 22 AA 0039 207 $SETBIT #NWA$V_RAC_FLG,(R7) ; Set first-time-thru flag for
20 0F 22 AA 0039 208 NT$CHK_RAC ;
20 0F 22 AA 0039 209 BSBB NT$CHK_SQO ; Reverse sense of SQO bit as required
20 0F 22 AA 0039 210 BBC #IFB$V_SQO,(R10),10$ ; Branch if not sequential-only
20 0F 22 AA 0039 211 $SETBIT #NWA$V_FTM_INIT,(R7) ; Set SEQ file transfer one-shot flag
20 0F 22 AA 0039 212 $CLRBIT #NWA$V_FTM_EOF,(R7) ; Initialize EOF flag
20 0F 22 AA 0039 213 10$: RMSSUC ; Return success
20 0F 22 AA 0039 214

```

NTOCONN  
V04-000

NETWORK CONNECT STREAM  
NT\$CONNECT - PERFORM NETWORK CONNECT FUN

N 12

15-SEP-1984 23:51:33 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:21 [RMS.SRC]NTOCONN.MAR;1

Page 6  
(3)

FF7F' 05 007D 214 EXIT: RSB  
31 007E 215 ERRSUP: BRW

NT\$RMT\_ROP

; Exit with RMS code in R0  
; Branch aid

```

0081 217 .SBTTL NT$CHK_SQO
0081 218
0081 219 ;++
0081 220 ; NT$CHK_SQO will override the user's setting of the SQO bit in the FOP field
0081 221 ; if any of the following conditions are met:
0081 222
0081 223 ; (1) IFB$V_SQO is set (forcing DAP file transfer mode) if RAC=SEQ, record
0081 224 ; access is requested and the remote node does not support DAP record
0081 225 ; access mode. This is useful for communicating with many non-VAX nodes
0081 226 ; through higher-level language programs that sequentially access remote
0081 227 ; files.
0081 228
0081 229 ; (2) IFB$V_SQO is set (forcing DAP file transfer mode) if the remote node
0081 230 ; does not support access by virtual block number (VBN) and the requested
0081 231 ; access is block I/O
0081 232
0081 233 ; (3) IFB$V_SQO is cleared (forcing DAP record access mode) if remote device
0081 234 ; is a terminal, the requested access is record access and the remote
0081 235 ; node supports DAP record access mode. This allows programs to perform
0081 236 ; bi-directional I/O to VMS terminals with SQO set. This is important to
0081 237 ; higher-level languages such as FORTRAN, whose run-time library sets
0081 238 ; SQO if RAC=SEQ.
0081 239
0081 240 ; (4) IFB$V_SQO is cleared in either record access or block mode access if
0081 241 ; the accessed file is an image indirect PPF, thus allowing successful
0081 242 ; completion of control operations, such as $DISPLAY.
0081 243 ;--
0081 244
0081 245 NT$CHK_SQO:: ; Entry point
10 22 05 E0 0081 246 BBS #IFB$V_BIO,- ; Branch if block I/O specified
0083 247 IFB$B_FAC(R10),20$
0086 248
0086 249 ;
0086 250 ; Enable DAP file transfer mode for record I/O if conditions in (1) are met.
0086 251 ;
0086 252
0086 253 BBS #DAP$V_SEQRAC,- ; Branch if remote node supports
0088 254 DAP$Q_SYSCAP(R7),30$ ; DAP record access mode
00 15 28 A7 E0 008B 255 CMPB RAB$B_RAC(R8),#RAB$C_SEQ ; Branch if access mode is not
00 1E A8 91 008F 256 BNEQ 10$ ; sequential
00 04 12 0091 257 $SETBIT #IFB$V_SQO,(R10) ; Internally set SQO, overriding user
0095 258 10$: RSB
0096 259
0096 260 ;
0096 261 ; Enable DAP file transfer mode for block I/O if conditions in (2) are met.
0096 262 ;
0096 263
0096 264 20$: BBS #DAP$V_RANVBN,- ; Branch if remote node supports random
0098 265 DAP$Q_SYSCAP(R7),40$ ; access by VBN
009B 266 $SETBIT #IFB$V_SQO,(R10) ; Internally set SQO, overriding user
009F 267 RSB
00A0 268
00A0 269 ;
00A0 270 ; Disable DAP file transfer mode if conditions in (3) or (4) are met.
00A0 271 ;
00A0 272
04 67 10 E0 00A0 273 30$: BBS #NWA$V_DEVTRM,(R7),50$ ; Leave the SQO bit alone if we are

```

NTOCONN  
V04-000

NETWORK CONNECT STREAM  
NT\$CHK\_SQ0

C 13

15-SEP-1984 23:51:33 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:21 [RMS.SRC] NTOCONN.MAR;1

Page 8  
(4)

|    |    |    |    |      |     |       |          |                                                                |
|----|----|----|----|------|-----|-------|----------|----------------------------------------------------------------|
| 04 | 6A | 22 | E1 | 00A4 | 274 | 40\$: | BBC      | #IFB\$V_PPF_IMAGE,(R10),60\$ ; not a terminal or an image PPF. |
|    |    |    |    | 00A8 | 275 | 50\$: | \$CLRBIT | #IFB\$V_SQ0,(R10) ; Internally clear SQ0, overriding user      |
|    |    |    | 05 | 00AC | 276 | 60\$: | RSB      |                                                                |

```

                                .SBTTL  NT$CHK_RAC
                                00AD  278
                                00AD  279
                                00AD  280 ;++
                                00AD  281 ; NT$CHK_RAC compares the current RMS RAC value with the previous RAC value used
                                00AD  282 ; to check for a switch in access mode.
                                00AD  283 ;--
                                00AD  284
                                00AD  285 NT$CHK_RAC::
OD 67 07 E4 00AD 286 BBSC #NWA$V_RAC_FLG,(R7),10$ ; Entry point
      1E A8 91 00B1 287 CMPB RAB$B_RAC(R8),- ; Check and clear first-time-thru flag
      00C8 C7 00B4 288 NWA$B_RMS_RAC(R7) ; Branch if no switch in access mode
              15 13 00B7 289 BEQL 20$ ;
              0C E1 00B9 290 BBC #DAP$V_SWMODE,- ; Branch if switching of access mode
      14 28 A7 00BB 291 DAP$Q_SYSCAP(R7),40$ ; is not supported by FAL
      1E A8 90 00BE 292 10$: MOVB RAB$B_RAC(R8),- ; Save user RAC value
      00C8 C7 00C1 293 NWA$B_RMS_RAC(R7) ;
              0F 10 00C4 294 BSBB GET_RAC ; Get DAP RAC value
      00C9 C7 08 50 E9 00C6 295 BLBC R0,30$ ; Branch on error
              51 90 00C9 296 MOVB R1,NWA$B_DAP_RAC(R7) ; Save DAP RAC value
              05 00CE 297 20$: RMSSUC ; Return success
              31 00D1 298 30$: RSB ; Exit
      FF2B' 00D2 299 40$: BRW NT$RMT_RAC ; Declare error and exit

```

```

00D5 301 :++
00D5 302 : GET_RAC returns the DAP_RAC value in R1 if supported by the remote FAL.
00D5 303 : The DAP value is determined by the combination of IFB$V_BIO, IFB$V_SQO,
00D5 304 : and RAB$B_RAC.
00D5 305 :--
00D5 306
00D5 307 GET_RAC:
00D5 308 RMSSUC ; Anticipate success
17 6A 2D E1 00D8 309 BBC #IFB$V_SQO,(R10),20$ ; Branch if not in file xfer mode
00 91 00DC 310 CMPB #RAB$C_SEQ,- ; Sequential access is the only one
1E A8 00DE 311 RAB$B_RAC(R8) ; allowed in FTM.
4E 12 00E0 312 BNEQ 90$ ; If not sequential return an error
00E2 313
51 05 9A 00E2 314 MOVZBL #DAP$K_BLK_FILE,R1 ; BIO=1; SQO=1; RAC=SEQ (assumed)
05 E0 00E5 315 BBS #IFB$V_BIO,- ; Determine if record or block seq.
03 22 AA 00E7 316 IFB$B_FAC(R10),10$
51 03 9A 00EA 317 MOVZBL #DAP$K_SEQ_FILE,R1 ; BIO=0; SQO=1; RAC=SEQ
05 E1 00ED 318 10$: BBC #DAP$V_SEQFIL,- ; Branch if not supported by FAL
3B 28 A7 00EF 319 DAP$Q_SYSCAP(R7),80$ ; Note: there is no BLKFIL bit defined
05 00F2 320 RSB ; (i.e., block I/O file xfer mode)
00F3 321
2C 22 AA 05 E0 00F3 322 20$: BBS #IFB$V_BIO,- ; Branch if block I/O specified
00F5 323 IFB$B_FAC(R10),60$
00F8 324 $CASEB SELECTOR=RAB$B_RAC(R8)- ; Access mode:
00F8 325 DISPL=<- ;
00F8 326 30$- ; Sequential access
00F8 327 40$- ; Random access by key value
00F8 328 50$- ; Random access by RFA
00F8 329 >
51 00 9A 0103 330 30$: MOVZBL #DAP$K_SEQ_ACC,R1 ; BIO=0; SQO=0; RAC=SEQ
21 E1 0106 331 BBC #DAP$V_SEQRAC,- ; Branch if not supported by FAL
22 28 A7 0108 332 DAP$Q_SYSCAP(R7),80$
05 010B 333 RSB
51 01 9A 010C 334 40$: MOVZBL #DAP$K_KEY_ACC,R1 ; BIO=0; SQO=0; RAC=KEY
08 E1 010F 335 BBC #DAP$V_RANKEY,- ; Branch if not supported by FAL
01 28 A7 0111 336 DAP$Q_SYSCAP(R7),45$ ; (random access by key for IDX org)
05 0114 337 RSB
13 28 A7 06 E1 0115 338 45$: BBC #DAP$V_RANRRN,- ; Branch if not supported by FAL
0117 339 DAP$Q_SYSCAP(R7),80$ ; (random access by RRN for REL org)
05 011A 340 RSB
51 02 9A 011B 341 50$: MOVZBL #DAP$K_RFA_ACC,R1 ; BIO=0; SQO=0; RAC=RFA
0A 0A E1 011E 342 BBC #DAP$V_RANRFA,- ; Branch if not supported by FAL
28 A7 0120 343 DAP$Q_SYSCAP(R7),80$
05 0123 344 RSB
51 04 9A 0124 345 60$: MOVZBL #DAP$K_BLK_VBN,R1 ; BIO=1; SQO=0; RAC=xxx
07 E1 0127 346 BBC #DAP$V_RANVBN,- ; Branch if not supported by FAL
01 28 A7 0129 347 DAP$Q_SYSCAP(R7),80$
05 012C 348 RSB
012D 349
FED0' 31 012D 350 80$: BRW NT$RMT_RAC ; Declare error and exit
0130 351 90$: RMSERR FTM ; Declare a File Transfer Mode error.
05 0135 352 RSB

```

```
0136 354 .SBTTL NT$MAP_ROP
0136 355
0136 356 :++
0136 357 : Encode the ROP field.
0136 358 :--
0136 359
51 04 A8 D0 0136 360 NT$MAP_ROP:: ; Entry point
52 D4 0136 361 MOVL RAB$L_ROP(R8),R1 ; Get ROP bits
013A 362 CLRL R2 ; Clear resultant ROP bits
013C 363
013C 364 :
013C 365 : The following DAP bits are defined in the DAP V4.1 specification.
013C 366 : These will be mapped.
013C 367 :
013C 368
013C 369 $MAPBIT RAB$V_EOF,DAP$V_EOF ; Map EOF bit
0144 370
0144 371 :
0144 372 : The following DAP bits are defined in the DAP V5.2 specification.
0144 373 : These will be mapped iff partner has implemented to DAP since V5.2 and for
0144 374 : UIF and TPT additional checks will be made.
0144 375 :
0144 376
3A 67 22 E1 0144 377 BBC #DAP$V_GEQ_V52,(R7),10$ ; Branch if partner uses DAP before V5.2
0148 378 $MAPBIT RAB$V_FDL,DAP$V_FDL ; Map FDL bit
0150 379 $MAPBIT RAB$V_LOA,DAP$V_LOA ; Map LOA bit
0158 380 $MAPBIT RAB$V_KGE,DAP$V_KGE ; Map KGE bit
0160 381 $MAPBIT RAB$V_KGT,DAP$V_KGT ; Map KGT bit
16 67 30 E1 0168 382 BBC #DAP$V_RMS,(R7),10$ ; Branch if partner is not RMS based
016C 383 $MAPBIT RAB$V_UIF,DAP$V_UIF ; Map UIF bit
06 A7 30 B3 0174 384 BITW #<<1a2DAP$V_VAXVMS-PARTNER>>!--
0178 385 <1a2DAP$V_VAXELAN-PARTNER>>!--
0178 386 0>,DAP$W_PARTNER(R7) ; Branch if partner is neither VAX/VMS
08 13 0178 387 BEQL 10$ ; nor VAXELAN
017A 388 $MAPBIT RAB$V_TPT,DAP$V_TPT ; Map TPT bit
0182 389 : ***** $MAPBIT RAB$V_RAH,DAP$V_RAH ; Map RAH bit
0182 390 : ***** $MAPBIT RAB$V_WBH,DAP$V_WBH ; Map WBH bit
0182 391
0182 392 :
0182 393 : The following DAP bits are defined in the DAP V5.4 specification.
0182 394 : These will be mapped iff partner has implemented to DAP since V5.4 and
0182 395 : partner is VAX/VMS.
0182 396 :
0182 397
26 67 23 E1 0182 398 10$: BBC #DAP$V_GEQ_V54,(R7),20$ ; Branch if partner uses DAP before V5.4
06 A7 30 B3 0186 399 BITW #<<1a2DAP$V_VAXVMS-PARTNER>>!--
018A 400 <1a2DAP$V_VAXELAN-PARTNER>>!--
018A 401 0>,DAP$W_PARTNER(R7) ; Branch if partner is neither VAX/VMS
20 13 018A 402 BEQL 20$ ; nor VAXELAN
018C 403 $MAPBIT RAB$V_ULK,DAP$V_ULK ; Map ULK bit
0194 404 $MAPBIT RAB$V_NLK,DAP$V_NLK ; Map NLK bit
019C 405 $MAPBIT RAB$V_RLK,DAP$V_RLK ; Map RLK bit
01A4 406 $MAPBIT RAB$V_BIO,DAP$V_ROPBIO ; Map BIO bit
01AC 407
01AC 408 :
01AC 409 : The following DAP bits are defined in the DAP V6.0 specification.
01AC 410 : These will be mapped iff partner has implemented to DAP since V6.0 and
```

```

01AC 411 ; (except for LIM) partner is VAX/VMS.
01AC 412 ;
01AC 413 ;
2C 67 25 E1 01AC 414 20$: BBC #DAP$V GEQ V60,(R7),30$ ; Branch if partner uses DAP before V6.0
01B0 415 $MAPBIT RAB$V LIM,DAP$V LIM ; Map LIM bit
20 67 34 E1 01B8 416 BBC #DAP$V VAXVMS,(R7),30$ ; Branch if partner is not VAX/VMS
01BC 417 $MAPBIT RAB$V_NXR,DAP$V_NXR ; Map NXR bit
01C4 418 $MAPBIT RAB$V_WAT,DAP$V_ROPWAT ; Map WAT bit
01CC 419 $MAPBIT RAB$V_RRL,DAP$V_RRL ; Map RRL bit
01D4 420 $MAPBIT RAB$V_REA,DAP$V_REA ; Map REA bit
51 52 D0 01DC 421 30$: MOVL R2,R1 ; Move data to correct register
FE1E' 30 01DF 422 BSBW NT$CVT_BN4_EXT ; Store ROP as an extensible field
05 01E2 423 RSB ; Exit
01E3 424
01E3 425 .END ; End of module

```

NTCONN  
Symbol table

NETWORK CONNECT STREAM

H 13

15-SEP-1984 23:51:33 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:21 [RMS.SRC]NTCONN.MAR;1

Page 13  
(7)

```

$$PSECT_EP      = 00000000
$$COUNT        = 00000003
$$RMSTEST       = 0000001A
$$RMS_PBUGCHK   = 00000010
$$RMS_TBUGCHK   = 00000008
$$RMS_UMODE     = 00000004
DAP$B_BITCNT    = 00000035
DAP$B_BLKCNT    = 00000056
DAP$B_CTLFUNC   = 00000040
DAP$B_DCODE_FID = 00000019
DAP$B_DCODE_MAC = 00000018
DAP$B_DCODE_MSG = 0000001A
DAP$B_DECVER    = 00000047
DAP$B_ECONUM    = 00000045
DAP$B_FILESYS   = 00000043
DAP$B_FLAGS     = 00000031
DAP$B_KRF       = 00000047
DAP$B_LEN256    = 00000034
DAP$B_LENGTH    = 00000033
DAP$B_OSTYPE    = 00000042
DAP$B_RAC       = 00000046
DAP$B_STREAMID  = 00000032
DAP$B_TYPE      = 00000030
DAP$B_USRNUM    = 00000046
DAP$B_USRVER    = 00000048
DAP$B_VERNUM    = 00000044
DAP$B_X_FIELD   = 00000024
DAP$C_BCN       = 000000C0
DAP$K_ACK_MSG   = 00000006
DAP$K_BLK_FILE  = 00000005
DAP$K_BLK_VBN   = 00000004
DAP$K_BLN       = 00C000C0
DAP$K_CONNECT   = 00000002
DAP$K_CTL_MSG   = 00000004
DAP$K_KEY_ACC   = 00000001
DAP$K_RFA_ACC   = 00000002
DAP$K_SEQ_ACC   = 00000000
DAP$K_SEQ_FILE  = 00000003
DAP$L_CMWA      = 00000030
DAP$L_CRC_RSLT  = 00000020
DAP$L_DCODE_STS = 00000018
DAP$L_MSG_MASK  = 0000001C
DAP$L_ROP       = 00000050
DAP$L_SSPWA     = 00000080
DAP$L_TEMP      = 00000090
DAP$M_BITCNT    = 00000008
DAP$M_BLKCNT    = 00000040
DAP$M_KRF       = 00000004
DAP$M_ROP       = 00000008
DAP$M_SEGMENT   = 00000040
DAP$M_TMP1$     = 00000008
DAP$M_TMP2$     = FFF80000
DAP$Q_DCODE_FLG = 00000000
DAP$Q_KEY       = 00000048
DAP$Q_MSG_BUF1  = 00000008
DAP$Q_MSG_BUF2  = 00000010
DAP$Q_SYSCAP    = 00000028

```

```

DAP$Q_SYSPEC    = 00000038
DAP$V_APPEND    = 0000000D
DAP$V_EOF       = 00000000
DAP$V_FDL       = 00000001
DAP$V_GEQ_V52   = 00000022
DAP$V_GEQ_V54   = 00000023
DAP$V_GEQ_V60   = 00000025
DAP$V_KGE       = 00000009
DAP$V_KGT       = 0000000A
DAP$V_LIM       = 0000000E
DAP$V_LOA       = 00000004
DAP$V_NLK       = 0000000B
DAP$V_NXR       = 0000000F
DAP$V_RANKEY    = 00000008
DAP$V_RANRFA    = 0000000A
DAP$V_RANRRN    = 00000006
DAP$V_RANVBN    = 00000007
DAP$V_REA       = 00000012
DAP$V_RLK       = 0000000C
DAP$V_RMS       = 00000030
DAP$V_ROPBIO    = 0000000D
DAP$V_ROPWAT    = 00000010
DAP$V_RRL       = 00000011
DAP$V_SEQFIL    = 00000005
DAP$V_SEQRAC    = 00000021
DAP$V_SWMODE    = 0000000C
DAP$V_TPT       = 00000006
DAP$V_UIF       = 00000002
DAP$V_ULK       = 00000005
DAP$V_VAXELAN   = 00000035
DAP$V_VAXVMS    = 00000034
DAP$W_BUFSIZ    = 00000040
DAP$W_CTLMENU   = 00000044
DAP$W_DISPLAY2  = 00000054
DAP$W_PARTNER   = 00000006
DAP$W_VERSION   = 00000004
ERRSUP          = 0000007E R 01
EXIT            = 0000007D R 01
GET_RAC         = 000000D5 R 01
IFB$B_FAC       = 00000022
IFB$B_NWA_PTR   = 0000003C
IFB$V_BIO       = 00000005
IFB$V_BRO       = 00000006
IFB$V_PPF_IMAGE = 00000022
IFB$V_SQO       = 0000002D
IRB$V_DAP_CONN  = 0000003C
NT$BUTLD_HEAD   = ***** X 01
NT$BUTLD_TAIL   = ***** X 01
NT$CHK_RAC      = 000000AD RG 01
NT$CHK_SQO      = 00000081 RG 01
NT$CONNECT      = 00000000 RG 01
NT$CVT_BN4_EXT  = ***** X 01
NT$MAP_ROP      = 00000136 RG 01
NT$RECEIVE      = ***** X 01
NT$RMT_RAC      = ***** X 01
NT$RMT_ROP      = ***** X 01
NT$TRANSMIT     = ***** X 01

```

NTCONN  
Symbol table

NETWORK CONNECT STREAM

I 13

15-SEP-1984 23:51:33 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:21 [RMS.SRC]NTCONN.MAR;1

Page 14  
(7)

NWASB\_ALLXABCNT 0000011C  
NWASB\_DAP\_RAC 000000C9  
NWASB\_FILESYS 000000C5  
NWASB\_KEYXABCNT 0000011D  
NWASB\_NETSTRSIZ 0000016F  
NWASB\_NODBUFSIZ 00000168  
NWASB\_ORG 000000C6  
NWASB\_OSTYPE 000000C4  
NWASB\_RFM 000000C7  
NWASB\_RMS\_RAC 000000C8  
NWASC\_BLN 00000800  
NWASK\_BLN 00000800  
NWASK\_IDX = 00000020  
NWASL\_ALLXABADR 00000100  
NWASL\_DATXABADR 00000104  
NWASL\_DEV 000000C0  
NWASL\_FHCXABADR 00000108  
NWASL\_KEYXABADR 0000010C  
NWASL\_MSG\_MASK 000000D4  
NWASL\_PROXABADR 00000110  
NWASL\_RDTXABADR 00000114  
NWASL\_SAVE\_FLGS 00000128  
NWASL\_SUMXABADR 00000118  
NWASL\_THREAD 000000FC  
NWASL\_XLTATTR 00000238  
NWASL\_XLTBUFFLG 0000022C  
NWASL\_XLTCNT 00000228  
NWASL\_XLTMAXINDEX 00000234  
NWASL\_XLTSIZ 00000230  
NWASQ\_ACS 00000244  
NWASQ\_BIGBUF 00000170  
NWASQ\_BLD 000000F0  
NWASQ\_FLG 00000000  
NWASQ\_INODE 0000025C  
NWASQ\_IOSB 000000D8  
NWASQ\_LNODE 00000160  
NWASQ\_LOGNAME 0000023C  
NWASQ\_NCB 00000264  
NWASQ\_RCV 000000E0  
NWASQ\_SAVE\_DESC 00000120  
NWASQ\_XLTBUF1 0000024C  
NWASQ\_XLTBUF2 00000254  
NWASQ\_XMT 000000E8  
NWASt\_ACSBUF 0000026C  
NWASt\_AUXBUF 000005E0  
NWASt\_DAP 00000000  
NWASt\_INODEBUF 000004AC  
NWASt\_ITM\_ATTR 00000200  
NWASt\_ITM\_END 00000224  
NWASt\_ITM\_LST 00000200  
NWASt\_ITM\_MAXINDEX 00000218  
NWASt\_ITM\_STRING 0000020C  
NWASt\_NCBBUF 0000052C  
NWASt\_NODEBUF 00000169  
NWASt\_RCVBUF 000001A0  
NWASt\_SCAN 00000100  
NWASt\_TEMP 00000120

NWASt\_XLTBUF1 000002AC  
NWASt\_XLTBUF2 000003AC  
NWASt\_XMTBUF 000003C0  
NWASt\_DEVTRM = 00000010  
NWASt\_FTM\_EOF = 0000001D  
NWASt\_FTM\_INIT = 00000019  
NWASt\_LAST\_MSG = 00000000  
NWASt\_RAC\_FLG = 00000007  
NWASt\_BUICD 000000D2  
NWASt\_DAPBUFSIZ 000000CA  
NWASt\_DIR\_OFF 000000CC  
NWASt\_DISPLAY 000000D0  
NWASt\_FIL\_OFF 000000CE  
NWASt\_JNLXABJOP 0000011E  
PARTNER = 00000030  
PIOSt\_TRACE \*\*\*\*\* X 01  
RABSt\_KRF = 00000035  
RABSt\_RAC = 0000001E  
RABSt\_SEQ = 00000000  
RABSt\_ROP = 00000004  
RABSt\_BIO = 0000000B  
RABSt\_EOF = 00000008  
RABSt\_FDL = 00000006  
RABSt\_KGE = 00000015  
RABSt\_KGT = 00000016  
RABSt\_LIM = 0000000E  
RABSt\_LOA = 0000000D  
RABSt\_NLK = 00000014  
RABSt\_NXR = 00000017  
RABSt\_REA = 00000002  
RABSt\_RLK = 00000013  
RABSt\_RRL = 00000003  
RABSt\_TPT = 00000001  
RABSt\_UIF = 00000004  
RABSt\_ULK = 00000012  
RABSt\_WAT = 00000011  
RECV\_ACK 00000059 R 01  
RMSt\_FTM = 000187C4  
SEND\_CTL 0000001F R 01  
TPTSt\_NTCONN \*\*\*\*\* X 01

+-----+  
! Psect synopsis !  
+-----+

| PSECT name  | Allocation        | PSECT No. | Attributes |
|-------------|-------------------|-----------|------------|
| . ABS       | 00000000 ( 0.)    | 00 ( 0.)  | NOPIC USR  |
| NF\$NETWORK | 000001E3 ( 483.)  | 01 ( 1.)  | PIC USR    |
| \$ABS\$     | 00000800 ( 2048.) | 02 ( 2.)  | NOPIC USR  |

+-----+  
! Performance indicators !  
+-----+

| Phase                  | Page faults | CPU Time    | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization         | 32          | 00:00:00.05 | 00:00:00.81  |
| Command processing     | 114         | 00:00:00.59 | 00:00:03.08  |
| Pass 1                 | 321         | 00:00:11.98 | 00:00:31.24  |
| Symbol table sort      | 0           | 00:00:01.63 | 00:00:02.72  |
| Pass 2                 | 90          | 00:00:02.14 | 00:00:07.92  |
| Symbol table output    | 25          | 00:00:00.27 | 00:00:01.04  |
| Psect synopsis output  | 1           | 00:00:00.02 | 00:00:00.07  |
| Cross-reference output | 0           | 00:00:00.00 | 00:00:00.00  |
| Assembler run totals   | 585         | 00:00:16.69 | 00:00:46.88  |

The working set limit was 1350 pages.  
62993 bytes (124 pages) of virtual memory were used to buffer the intermediate code.  
There were 60 pages of symbol table space allocated to hold 1108 non-local and 52 local symbols.  
425 source lines were read in Pass 1, producing 14 object records in Pass 2.  
28 pages of virtual memory were used to define 27 macros.

+-----+  
! Macro library statistics !  
+-----+

| Macro library name                  | Macros defined |
|-------------------------------------|----------------|
| -\$255\$DUA28:[RMS.OBJ]RMS.MLB;1    | 19             |
| -\$255\$DUA28:[SYSLIB]STARLET.MLB;2 | 4              |
| TOTALS (all libraries)              | 23             |

1343 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:NTOCONN/OBJ=OBJ\$:NTOCONN MSRC\$:NTOCONN/UPDATE=(ENH\$:NTOCONN)+LIB\$:RMS/LIB

0315 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

NT0ACCESS  
LIS

NT0CLOSE  
LIS

NT0BLDXAB  
LIS

NT0CONN  
LIS

NT0CREATE  
LIS

NT0DAP10  
LIS

NT0DAPCRC  
LIS

NT0ACCFIL  
LIS

NT0BLK10  
LIS